**Docker compose process**

[**https://github.com/docker/compose/blob/release/docs/gettingstarted.md**](https://github.com/docker/compose/blob/release/docs/gettingstarted.md)

**Getting Started**

On this page you build a simple Python web application running on Docker Compose. The application uses the Flask framework and increments a value in Redis. While the sample uses Python, the concepts demonstrated here should be understandable even if you're not familiar with it.

**Prerequisites**

Make sure you have already [installed both Docker Engine and Docker Compose](https://github.com/docker/compose/blob/release/docs/install.md). You don't need to install Python, it is provided by a Docker image.

**Step 1: Setup**

1. Create a directory for the project:
2. $ mkdir composetest
3. $ cd composetest
4. With your favorite text editor create a file called app.py in your project directory.
5. from flask import Flask
6. from redis import Redis
7. app = Flask(\_\_name\_\_)
8. redis = Redis(host='redis', port=6379)
9. @app.route('/')
10. def hello():
11. redis.incr('hits')
12. return 'Hello World! I have been seen %s times.' % redis.get('hits')
13. if \_\_name\_\_ == "\_\_main\_\_":
14. app.run(host="0.0.0.0", debug=True)
15. Create another file called requirements.txt in your project directory and add the following:
16. flask
17. redis

These define the applications dependencies.

**Step 2: Create a Docker image**

In this step, you build a new Docker image. The image contains all the dependencies the Python application requires, including Python itself.

1. In your project directory create a file named Dockerfile and add the following:
2. FROM python:2.7
3. ADD . /code
4. WORKDIR /code
5. RUN pip install -r requirements.txt
6. CMD python app.py

This tells Docker to:

* + Build an image starting with the Python 2.7 image.
  + Add the current directory . into the path /code in the image.
  + Set the working directory to /code.
  + Install the Python dependencies.
  + Set the default command for the container to python app.py

For more information on how to write Dockerfiles, see the [Docker user guide](https://docs.docker.com/engine/userguide/dockerimages/#building-an-image-from-a-dockerfile) and the [Dockerfile reference](http://docs.docker.com/reference/builder/).

1. Build the image.
2. $ docker build -t web .

This command builds an image named web from the contents of the current directory. The command automatically locates the Dockerfile, app.py, and requirements.txt files.

**Step 3: Define services**

Define a set of services using docker-compose.yml:

1. Create a file called docker-compose.yml in your project directory and add the following:
2. version: '2'
3. services:
4. web:
5. build: .
6. ports:
7. - "5000:5000"
8. volumes:
9. - .:/code
10. depends\_on:
11. - redis
12. redis:
13. image: redis

This Compose file defines two services, web and redis. The web service:

* Builds from the Dockerfile in the current directory.
* Forwards the exposed port 5000 on the container to port 5000 on the host machine.
* Mounts the project directory on the host to /code inside the container allowing you to modify the code without having to rebuild the image.
* Links the web service to the Redis service.

The redis service uses the latest public [Redis](https://registry.hub.docker.com/_/redis/) image pulled from the Docker Hub registry.

**Step 4: Build and run your app with Compose**

1. From your project directory, start up your application.
2. $ docker-compose up
3. Pulling image redis...
4. Building web...
5. Starting composetest\_redis\_1...
6. Starting composetest\_web\_1...
7. redis\_1 | [8] 02 Jan 18:43:35.576 # Server started, Redis version 2.8.3
8. web\_1 | \* Running on http://0.0.0.0:5000/
9. web\_1 | \* Restarting with stat

Compose pulls a Redis image, builds an image for your code, and start the services you defined.

1. Enter http://0.0.0.0:5000/ in a browser to see the application running.

If you're using Docker on Linux natively, then the web app should now be listening on port 5000 on your Docker daemon host. If [http://0.0.0.0:5000](http://0.0.0.0:5000/) doesn't resolve, you can also try http://localhost:5000.

If you're using Docker Machine on a Mac, use docker-machine ip MACHINE\_VM to get the IP address of your Docker host. Then, open http://MACHINE\_VM\_IP:5000 in a browser.

You should see a message in your browser saying:

Hello World! I have been seen 1 times.

1. Refresh the page.

The number should increment.

**Step 5: Experiment with some other commands**

If you want to run your services in the background, you can pass the -d flag (for "detached" mode) to docker-compose upand use docker-compose ps to see what is currently running:

$ docker-compose up -d

Starting composetest\_redis\_1...

Starting composetest\_web\_1...

$ docker-compose ps

Name Command State Ports

-------------------------------------------------------------------

composetest\_redis\_1 /usr/local/bin/run Up

composetest\_web\_1 /bin/sh -c python app.py Up 5000->5000/tcp

The docker-compose run command allows you to run one-off commands for your services. For example, to see what environment variables are available to the web service:

$ docker-compose run web env

See docker-compose --help to see other available commands. You can also install [command completion](https://github.com/docker/compose/blob/release/docs/completion.md) for the bash and zsh shell, which will also show you available commands.

If you started Compose with docker-compose up -d, you'll probably want to stop your services once you've finished with them:

$ docker-compose stop

At this point, you have seen the basics of how Compose works.

**Where to go next**

* Next, try the quick start guide for [Django](https://github.com/docker/compose/blob/release/docs/django.md), [Rails](https://github.com/docker/compose/blob/release/docs/rails.md), or [WordPress](https://github.com/docker/compose/blob/release/docs/wordpress.md).
* [Explore the full list of Compose commands](https://github.com/docker/compose/blob/release/docs/reference/index.md)
* [Compose configuration file reference](https://github.com/docker/compose/blob/release/docs/compose-file.md)